# Digital Design and Computer Architecture (252-0028-00L), Spring 2020
# Optional HW 1: DRAM Refresh and Combinational Logic

Instructor: Prof. Onur Mutlu
TAs: Mohammed Alser, Rahul Bera, Can Firtina, Juan Gomez-Luna, Jawad Haj-Yahya, Hasan Hassan, Konstantinos Kanellopoulos, Lois Orosa, Jisung Park, Geraldo De Oliveira Junior, Minesh Patel, Giray Yaglikci

Released: Monday, March 2, 2020

## 1 DRAM Refresh

A new supercomputer has a DRAM-based memory system with the following configuration:

- The total capacity is 1 ExaByte (EB).
- The DRAM row size is 8 KiloByte (KB).
- The minimum retention time among all DRAM rows in the system is 64 ms. In order to ensure that no data is lost, every DRAM row is refreshed once every 64 ms. (Note: For each calculation in this question, you may leave your answer in simplified form in terms of powers of 2 and powers of 10.)

(a) How many DRAM rows does the memory system have?

(b) How many DRAM refreshes happen in 64ms?

(c) What is the power consumption of DRAM refresh? (Hint: you will need to figure out how much current the DRAM device draws during refresh operations. You can find useful information in the technical note by Micron `https://safari.ethz.ch/digitaltechnik/spring2020/lib/exe/fetch.php?media=tn4704.pdf`. Use the current (IDD) numbers specified in the datasheet posted on the website `https://safari.ethz.ch/digitaltechnik/spring2020/lib/exe/fetch.php?media=1gb_ddr3_sdram.pdf`. Clearly state all the assumptions and show how you derive the power numbers. You are welcome to use other data sheets as well. Make sure you specify how you obtain the power numbers and show your calculations and thought process.)

(d) What is the total energy consumption of DRAM refresh during a refresh cycle? And during a day?

This question is an extended version of the question on Slide 79 in Lecture 2b: `https://safari.ethz.ch/digitaltechnik/spring2020/lib/exe/fetch.php?media=onur-digitaldesign-2020-lecture2b-mysteries-afterlecture.pdf`.
Video: `https://www.youtube.com/watch?v=3YnSjuhMhS0`.

## 2 Main Memory Potpourri

A machine has a 4 GB DRAM main memory system. Each row is refreshed every 64 ms.

(a) The machine's designer runs two applications A and B (each run alone) on the machine. Although applications A and B have a similar number of memory requests, application A spends a surprisingly larger fraction of cycles stalling for memory than application B does? What might be the reasons for this?

(b) Application A also consumes a much larger amount of memory energy than application B does. What might be the reasons for this?

(c) When applications A and B are run together on the machine, application A's performance degrades significantly, while application B's performance does not degrade as much. Why might this happen?

(d) The designer decides to use a smarter policy to refresh the memory. A row is refreshed only if it has not been accessed in the past 64 ms. Do you think this is a good idea? Why or why not?

(e) When this new refresh policy is applied, the refresh energy consumption drops significantly during a run of application B. In contrast, during a run of application A, the refresh energy consumption reduces only slightly. Is this possible? Why or why not?

## 3 Transistor-Level Circuit Design

In Lecture 4, we learned how to implement digital circuits using the CMOS technology (i.e., p-type and n-type MOS transistors). In this assignment, we ask you to schematically design circuits using CMOS transistors for the following logic gates:

- Exclusive OR Gate (XOR)

- Exclusive NOT OR Gate (XNOR)

## 4 Multiplexer (MUX)

Draw the following schematics for an 8-input (8:1) MUX.

- Gate level: as a combination of basic AND, OR, NOT gates. Use as few gates as possible.

- Module level: as a combination of 2-input (2:1) MUXes. Use as few 2-input MUXes as possible.

## 5 Logical Completeness

The set of {AND, OR, and NOT} gates is logically complete. We can build a circuit to carry out the specification of any truth table we wish, without using any other kind of gate. From Lecture 4, you know that the NOR gate by itself is also logically complete. Prove that you can build a circuit to carry out the specification of any truth table, by using only NOR gates.

## 6 Boolean Logic and Truth Tables

In this question we ask you to derive the boolean equations for two 4-input logic functions, $X$ and $Y$. Please use the truth table below to answer the following three questions.

| **Inputs** | | | | **Outputs** | |
|---|---|---|---|---|---|
| $A_3$ | $A_2$ | $A_1$ | $A_0$ | $X$ | $Y$ |
| 0 | 0 | 0 | 0 | | |
| 0 | 0 | 0 | 1 | | |
| 0 | 0 | 1 | 0 | | |
| 0 | 0 | 1 | 1 | | |
| 0 | 1 | 0 | 0 | | |
| 0 | 1 | 0 | 1 | | |
| 0 | 1 | 1 | 0 | | |
| 0 | 1 | 1 | 1 | | |
| 1 | 0 | 0 | 0 | | |
| 1 | 0 | 0 | 1 | | |
| 1 | 0 | 1 | 0 | | |
| 1 | 0 | 1 | 1 | | |
| 1 | 1 | 0 | 0 | | |
| 1 | 1 | 0 | 1 | | |
| 1 | 1 | 1 | 0 | | |
| 1 | 1 | 1 | 1 | | |

(a) The output $X$ is *one* when the input does **not** contain 3 consecutive 1's in the word $A_3, A_2, A_1, A_0$. The output $X$ is *zero*, otherwise. **Fill in the truth table above** and use the *product of sums* form to **write the corresponding boolean equation** for $X$. (*No simplification needed.*)

(b) The output $Y$ is *one* when no two adjacent bits in the word $A_3, A_2, A_1, A_0$ are the same (e.g., if $A_2$ is 0 then $A_3$ and $A_1$ cannot be 0). The output $Y$ is *zero*, otherwise (e.g., 0000). **Fill in the truth table above** and use the *sum of products* form to **write the corresponding boolean equation** for $Y$. (*No simplification needed.*)

(c) Please represent the circuit of $Y$ using *only* 2-input XOR and AND gates.